

# UTILITY PATENT APPLICATION TRANSMITTAL (Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.  
EN999088

Total Pages in this Submission  
3

## TO THE ASSISTANT COMMISSIONER FOR PATENTS

Box Patent Application  
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

**INFORMATION GATHERING FACILITY EMPLOYING DICTIONARY  
FILE CONTAINING MULTIPLE INQUIRIES**

and invented by:

**Douglas G. Murray**

If a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: \_\_\_\_\_

Enclosed are:

### Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 51 pages and including the following:
  - a. ☒ Descriptive Title of the Invention
  - b. ☒ Cross References to Related Applications (if applicable)
  - c. ☐ Statement Regarding Federally-sponsored Research/Development (if applicable)
  - d. ☐ Reference to Microfiche Appendix (if applicable)
  - e. ☒ Background of the Invention
  - f. ☒ Brief Summary of the Invention
  - g. ☒ Brief Description of the Drawings (if drawings filed)
  - h. ☒ Detailed Description
  - i. ☒ Claim(s) as Classified Below
  - j. ☒ Abstract of the Disclosure

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)**

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

Docket No.  
**EN999088**

Total Pages in this Submission  
**3**

**Application Elements (Continued)**

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*

- a. ☒ Formal                      Number of Sheets 8
- b. ☐ Informal                      Number of Sheets \_\_\_\_\_

4. ☒ Oath or Declaration

- a. ☒ Newly executed *(original or copy)*                      ☐ Unexecuted
- b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
- c. ☒ With Power of Attorney                      ☐ Without Power of Attorney
- d. ☐ DELETION OF INVENTOR(S)

Signed statement attached deleting inventor(s) named in the prior application,  
see 37 C.F.R. 1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*

The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Computer Program in Microfiche *(Appendix)*

7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*

- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy *(identical to computer copy)*
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

**Accompanying Application Parts**

8. ☒ Assignment Papers *(cover sheet & document(s))*

9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*

10. ☐ English Translation Document *(if applicable)*

11. ☒ Information Disclosure Statement/PTO-1449                      ☒ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

☐ First Class                      ☒ Express Mail *(Specify Label No.):* EL172581546US

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Large Entity)**

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

Docket No.  
EN999088

Total Pages in this Submission  
3

**Accompanying Application Parts (Continued)**

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☐ Additional Enclosures *(please identify below):*

**Fee Calculation and Transmittal**


**CLAIMS AS FILED**

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	70	- 20 =	50	x \$18.00	\$900.00
Indep. Claims	8	- 3 =	5	x \$78.00	\$390.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$760.00
OTHER FEE (specify purpose)					\$0.00
TOTAL FILING FEE					\$2,050.00

- ☐ A check in the amount of \_\_\_\_\_ to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. 09-0457 (IBM) as described below. A duplicate copy of this sheet is enclosed.
- ☒ Charge the amount of \$2,050.00 as filing fee.
  - ☒ Credit any overpayment.
  - ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
  - ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: 11/18/99

CC:

  
Signature  
John R. Pivnichny, Ph.D.  
Registration No. 43,001  
IBM Corporation  
Intellectual Property Law  
1701 North Street, N50/040-4  
Endicott, New York 13760  
Telephone (607) 755-6565  
Facsimile (607) 755-3250

11/18/99

Title: INFORMATION GATHERING FACILITY EMPLOYING  
DICTIONARY FILE CONTAINING MULTIPLE INQUIRIES

"EXPRESS MAIL" MAILING LABEL NO. EL172581546US

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and addressed to Assistant Commissioner for Patents, Box PATENT APPLICATION, Washington, D.C. 20231.

Denise M. Jurik

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

**APPLICATION  
FOR  
UNITED STATES LETTERS PATENT**

**APPLICANT(S) NAME:** Douglas George Murray

**TITLE:** INFORMATION GATHERING FACILITY EMPLOYING  
DICTIONARY FILE CONTAINING MULTIPLE  
INQUIRIES

**DOCKET NO.** EN999088

**INTERNATIONAL BUSINESS MACHINES CORPORATION**

**Certificate of Mailing Under 37 CFR 1.10**

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C., 20231 as "Express Mail Post Office to Addressee".

"Express Mail" Label Number EL172581546US

On 11/18/99

Denise M. Jurik

*Typed or Printed Name of Person Mailing Correspondence*

Denise M Jurik  
*Signature of Person Mailing Correspondence*

11/18/99  
*Date*

**INFORMATION GATHERING FACILITY EMPLOYING  
DICTIONARY FILE CONTAINING MULTIPLE INQUIRIES**

**Cross-Reference to Related Application**

5 This application contains subject matter which is  
related to the subject matter of the following application,  
which is assigned to the same Assignee as this application  
and which is hereby incorporated herein by reference in its  
entirety:

10 "Information Gathering Facility Employing Dictionary  
File Containing Inquiry Invoking An External Process," by  
Douglas G. Murray, (Docket No. EN999112), Serial No.  
\_\_\_\_\_, co-filed herewith.

**Technical Field**

15 The present invention relates in general to information  
processing within a computer system, and more particularly,  
to an information gathering facility, for example, for a  
network of computer systems for use in ascertaining updated  
state information on one or more computer system within the  
20 network. Still more particularly, the invention relates to  
a configuration of a dictionary file data structure employed  
in ascertaining state information on one or more computer  
systems of a network and to a technique for gathering such  
information using the dictionary file.

25 **Background of the Invention**

Computer networks provide a number of advantages over  
standalone systems. In a network, each computer is a node  
which communicates with other nodes over one or more links.  
Nodes may be provided which store and manage databases or

other data files on mass storage devices, or which manage printers or links to public telecommunication networks, etc.

As networks become larger, and more computers, input/output devices and other machines are connected  
5 together, management of the network becomes more and more difficult. To alleviate data transfer limitations when connecting many nodes to a single network, networks are often divided into a smaller number of nodes, essentially creating separate networks which are then interconnected by  
10 means of bridges or gateways to allow a node on one sub-network to communicate with a node on another sub-network. This alleviates the data transfer limitations of networks, but does not alleviate management problems.

One problem associated with management of a computer  
15 system is managing software on the different computing nodes. Software management can include a number of functions, such as verifying that a computing node has a particular type of file, a particular version of the file, and possibly, ascertaining contents of the file. Keeping  
20 track of software distribution and use can be important for licensing purposes, as well as for basic functioning of the network. In the past, when computer systems were a single large, multiuser system, a system manager performed these functions. With the advent of single user systems, such as  
25 personal computers and workstations, the users have essentially become the system manager, requiring them to perform these system management tasks individually.

A need thus exists in the art for an enhanced, automated information gathering facility which allows state  
30 data, such as file existence and file configuration

information, for one or more nodes to be maintained up-to-date in a control information repository of the computer system network.

### Disclosure of the Invention

5 Briefly summarized, the invention comprises in one aspect a method for gathering information on a state of a computer system. The method includes: providing a dictionary file having a plurality of inquiries for  
10 ascertaining state information on the computer system, the plurality of inquiries being organized into at least one subject group, each subject group being directed to a different piece of the state information, at least one group of the at least one subject group having multiple records of inquiry; and processing at least one inquiry of the  
15 plurality of inquiries of the dictionary file to accumulate the state information, the processing including for each group of the at least one group having multiple records of inquiry, processing a record of the multiple records of inquiry, and if a condition of the record is satisfied then  
20 terminating processing of the group, otherwise processing a next record of the multiple records of inquiry and continuing until a condition of one record of the multiple records of inquiry is satisfied or until all records of the multiple records of inquiry of the group have been  
25 processed.

In another aspect, a method is provided for gathering information on a state of a computer system which includes: providing a dictionary file having a plurality of inquiries for ascertaining state information on the computer system,  
30 at least one inquiry of the plurality of inquiries includes



an instruction having a result which is output when a condition of the instruction is satisfied; and processing the at least one inquiry of the plurality of inquiries of the dictionary file to accumulate the state information, the  
5 processing including for each instruction outputting the result of the instruction from the dictionary file when the condition of the instruction is satisfied, wherein the state information on the computer system includes outputted results from satisfaction of the at least one instruction.

10 Systems and program storage devices corresponding to the above-summarized methods, as well as data structures encompassing the dictionary files employed thereby, are also described and claimed herein.

To restate, presented herein is an information  
15 gathering facility which can reside at a central location within a network of computer systems and be read from the central location by the individual computer systems when state information is to be derived. Thus, there is the ability to change one or more inquiries at the central  
20 location without making changes at all computer systems of the network. In accordance with the principles of the present invention, there is also an ability to design and implement inquiries of complicated machine configurations without programming, i.e., creating an executable of the  
25 file. Further, the format of the inquiry results, that is the state information returned, can be defined in the dictionary file itself, i.e., each instruction or line of inquiry in the dictionary file can possess a condition which if satisfied outputs a result and the cumulative results

comprise the state information. This feature can be employed by other processes to query a machine and have the response formatted in a way that the process needs.

By providing multiple inquiries within a group,  
5 searching is possible for one or more of several versions of an application installed on a workstation in the network. When an inquiry finds the desired version on a workstation, the remaining inquiries are omitted. As new versions of an application become available, a new inquiry is simply added  
10 to the group that is looking for that application. This leaves intact all the inquiries that looked for the 'backlevel' versions. Code reuse is thus encouraged since other applications can use this tool to gather workstation information instead of including separate logic to do their  
15 inquiry.

#### **Brief Description of the Drawings**

The above-described objects, advantages and features of the present invention, as well as others, will be more  
20 readily understood from the following detailed description of certain preferred embodiments of the invention, when considered in conjunction with the accompanying drawings in which:

FIG. 1 depicts a computer system network implementing  
25 an information gathering facility in accordance with the principles of the present invention;

FIG. 2 depicts an example of possible types of inquiry which could be employed in a directory file in accordance with the principles of the present invention;

FIGS. 3A & 3B are a flowchart of one embodiment for processing a directory file implemented in accordance with the principles of the present invention;

FIG. 4 is a flowchart of one embodiment of a process  
5 file inquiry routine called by the directory file process flowchart of FIGS. 3A & 3B;

FIG. 5 is a flowchart of one embodiment of a process  
INI inquiry routine called by the directory file process flowchart of FIGS. 3A & 3B;

10 FIG. 6 is a flowchart of one embodiment of a process external process inquiry routine called by the dictionary file process of FIGS. 3A & 3B;

FIG. 7 is a flowchart of one embodiment of a default routine called by the directory process flowchart of FIGS.  
15 3A & 3B; and

FIG. 8 is an example of a dictionary file data structure implemented in accordance with the principles of the present invention.

#### **Best Mode for Carrying Out the Invention**

20 Generally stated, provided herein is an information gathering facility which employs a dictionary file having an unique data structure. Specifically, the dictionary file has a plurality of inquiries for ascertaining state information on a computer system within a network of  
25 computers. The plurality of inquiries are organized into one or more subject groups. One or more of the subject

groups contain multiple records or lines of inquiry. The information gathering technique employs this dictionary file by processing the inquiries of the subject groups to derive information on the state of the computer system. When

5 multiple records or lines of inquiry exist within a given group, processing terminates with a first record of inquiry which has a condition that is satisfied by the computer system. One or more of the plurality of inquiries within the dictionary file are referred to herein as "instructions"

10 and comprise inquiries with conditions that if satisfied cause the outputting of a result which is to be added to a file being constructed with the state information on the computer system. This feature allows programming (from the dictionary file) of the result to be output upon

15 satisfaction of an inquiry contained within the dictionary file. Thus, the form of the information returned as the state information is dictated from the dictionary file itself. In one embodiment, the dictionary file comprises an ASCII file which is maintained at a central location within

20 a computing network, and which is called by an inquiry tool (i.e., routine) resident on the computer system to be examined.

FIG. 1 depicts one embodiment of a computer network, generally denoted 10, which includes multiple computing

25 systems such as client workstations 12 coupled to a server 14. As shown, server 14 functions as an information repository in accordance with the principles of the present invention. Each client workstation 12 includes an inquiry tool 16 and in this example a copy of the dictionary file 18

30 read from information repository 14. Inquiry tool 16 includes the control file which tells, for example, client 12 when to update its corresponding state information, where

to go to obtain the dictionary file and where to return the  
state information 20. The inquiry tool is an application  
that is installed on a workstation that is used to gather  
information from the workstation. Once operational, the  
5 tool will gather the information when executed by the end  
user or optionally on a periodic schedule. Installation of  
this tool could be done at any time, but is typically done  
prior to a user receiving the workstation for use.

In this embodiment, the returned information 20 is  
10 stored at database 20 of information repository 14 of  
network 10. Preferably, the dictionary file copy at client  
12 is updated each time inquiry tool 16 initiates processing  
to update the client state information. Note that as used  
herein the "state of the computer system" and "state  
15 information" refer to, for example, whether a program or  
file exists on the client, the version or level of the  
program or file, or to a variable or value from an  
application residing on the client or resulting from  
execution of an external process by the client as described  
20 in greater detail in the above-incorporated, co-filed  
application.

The information gathering facility of the present  
invention can be employed with many types of computer system  
environments and many types of computer systems. For  
25 instance, the computer environment might include an RS/6000  
computer system running an AIX Operating System (RS/6000 and  
AIX are offered by International Business Machines  
Corporation). Alternatively, the computer environment could  
include a UNIX workstation running a UNIX-based operating  
30 system. Instead of a single system, the computing  
environment could comprise a network of computer systems

such as depicted in FIG. 1 wherein each computer system could be a UNIX or AIX workstation and the units are coupled to one another via a TCP/IP connection. Each unit might include, for example, a central processing unit, memory and one or more input/output devices which are well-known in the art. Again, the information gathering facility of the present invention can be incorporated and used with any type of computing unit, computer, processor, node, system, workstation and/or environment without departing from the spirit of the present invention. As another example, each computer system might comprise a PS/2 computer offered by International Business Machines Corporation, or one or more of the units may be based upon the Enterprise Systems Architecture offered by International Business Machines Corporation. Additionally, connection between the computer systems need not be TCP/IP. It can be any type of wire connection, token ring or network connection, to name just a few examples.

FIG. 2 depicts one example of several possible inquiry types which can be employed by a dictionary file constructed in accordance with the principles of the present invention. These inquiry types include: a file inquiry with checks for existence of a file of a certain date, time and/or size, and which can return file information; an INI file inquiry which checks for a certain application, variable and/or value, and which can return a certain value or all variables and values; an ASCII file inquiry which checks for a certain character string in a file, and which can return information on a line within the character string; a registry inquiry which checks for a certain registry tree and value, and which can return a value or all values within a tree or sub-tree; an external process inquiry using an INI output which

executes an external process and performs an INI file inquiry on the results; an external process inquiry using an ASCII output, which executes an external process and performs an ASCII file inquiry on the results; an external  
5 process inquiry using a registry which executes an external process and does a registry inquiry on the result; and a multiple inquiry which can comprise a combination of selected inquiry types wherein all inquiries must succeed.

By way of example, if application APP1 has an INI file  
10 entry of Version=1.0 for version 1.0 but for some reason the INI file is not used at all for version 2.0 but instead a new file APP1.DAT is used, the following would detect the installed version:

- 15 1. Check for APP1.DAT file on the system, if found version 2.0 is installed.
2. Check the INI file for Version=1.0 line, if found version 1.0 is installed.
3. Check for APP1.EXE to see if any version is installed.

20 As noted, one feature of an information gathering facility or Client Information Gathering Facility (CIGF) in accordance with the principles of the present invention is the dictionary file. This file is used to determine what the state of the computer system is and what information is  
25 to be returned to the server to be placed into the data repository. Within the dictionary file there are sections or groups of inquiries. Each group has one or more lines or records of inquiries that instruct CIGF how to gather the

desired state information for that particular subject group. This information can be, for example, a version or setting data as supported by the options described below. Within the subject groups, any or all lines of inquiry may be used  
5 to obtain a result. In one example, a "Notes" grouping may have two lines which when processed determine the output, if any.

```
10      [Notes]
      ;file;Notes:V4.5;notes.exe;3/11/1997;231936
      ;file;Notes:Unknown %fileid% %date% %time% %size%;notes.exe
      [next app]
      ;file;%appid%;V6;napp.exe;6/22/1996
```

First the file notes.exe is searched for in the file system. If it is found and is dated 03/11/1997 and its size  
15 is 231936 then the string Notes:V4.5 is output. Otherwise, the process continues for each line under the group until one of the lines is successful or there are no more lines for the group. In this example, if the Notes.exe file could not be found according to the first line, then the second  
20 line is processed. Since there is no date, time or size specified for the second line then it will succeed if the Notes.exe file exists anywhere on the system. The output line will contain the full path where it was found and the date, time and size of the file.

25 There are several options that are available to be used on the lines for each group. The first character on the line can be used as the delimiter for each of the fields used when an option is used to return information.

The following is a description of certain possible  
30 options and the syntax of the lines that implement them. Each option below is available on various operating systems,



such as the OS/2 Operating System offered by International Business Machines Corp.

### Check for Existence of a File

5 This option can be used to check for the existence of a file to determine the version of software that is installed. The file option can be used to check for the existence of a set of files and return a version statement if the files are found.

10 Syntax:  
;file;output;fileid1;date1;time1;size1;...;fileidn;daten;timen;sizen

### Output

15 This is the output that will be placed in the output file if the files are found. There are several variables that will be substituted into this string before it is returned.

%appid%

The current group or application id.

%del%

The input field delimiter.

20 %fileid%

The full path of the last file being searched for.

%date%

The date of the last file being searched for.

%time%

The time of the last file being searched for.

%size%

The size of the last file being searched for.

5 fileid1

The name and ext of a file to search for. This file is one that if it exists, the application is installed.

date1

10 The date that the file must have in order to be this version. \* or no value means that the date of the file is not important.

time1

15 The time that the file must have in order to be this version. \* or no value means that the time is not important.

size1

The size that the file must have in order to be this version. \* or no value means that the size is not important.

fileidn

20 Used if more than one file is to be checked.

daten

Used if more than one file is to be checked.

## timen

Used if more than one file is to be checked.

## size

Used if more than one file is to be checked.

## 5 Obtain Data from Execution of External Application

IPConfig obtains TCP/IP information from the system by executing the WINIPCFG command on Windows 95 or the IPCONFIG command on Windows NT. Use the IPConfig option to run these  
10 utilities and extract information from the result to be sent to the repository.

### Syntax:

;IPConfig;output;key;value

### Output

15 This is the output that will be placed in the output file if the key is found. There are several variables that will be substituted into this string before it is returned.

%appid%

The current application id.

20 %del%

The input field delimiter.

%key%

The Key being searched for.

%value%

The values found for all occurrences of the key within the app. When there are more than one value the delimiter is used between them.

## 5 key

This is an ASCII string of characters that are searched for at the beginning of each line output by the WINIPCFG or IPCONFIG commands. If found then the remainder of the line, following the : delimiter is used as the value associated with the key being searched for.

## value

If a value is given then the inquiry will succeed only if the value given matches that on the line of the WINIPCFG or IPCONFIG output. When \* is the value given or no value is given then the inquiry succeeds if the key is found in the WINIPCFG or IPCONFIG output and the value in that output is returned.

## Obtain Data from an ASCII INI File

Often application INI files contain information that is required to determine how that application is installed. Use the INI option to extract information from an application INI file. Those skilled in the art will note that an INI file is an initialization file in Windows which provides persistent storage of configuration data. Many applications employ INI files to store information that must persist. Those skilled in the art of Windows programming understand the format and use of INI files.

Syntax:

;ini;output;fileid;app;key;value;reg\_exe

## Output

This is the output that will be placed in the output file if the key is found. There are several variables that will be substituted into this string before it is returned.

5   %appid%

The current application (i.e., group) id.

%del%

The input field delimiter.

10   %fileid%

The full path of the INI file on the system.

%app%

The application from the INI file.

%key%

15   The key from the INI file.

%value%

20   The values found for all accuracies of the key within the application are placed here. When there are more than one value the delimiter is used between them.

## fileid

The name and extension of the INI file. A path may also be included but a drive letter is not supported.

## app

The application name within the INI file. This is required and is used to find the application in the file.

## 5 key

This is the key in the INI file. \* or no value means that the "n" lines are to be read from the application and placed in the output as a delimited list.

## 10 value

If a value is given then the version is defined by this value assigned to the key. \* or no value means that any value assigned to the key is all that is needed. When \* is given as the key, then the

15 number of lines to include in the output is given as the value. \* or no value in this case means include all lines from the application.

## reg\_exe (Windows NT/2000/98/95 Versions)

20 This is the name of the registered exe file that will use the INI file being searched. If reg\_exe is given then the registered install directory for the executable is first searched for the INI file and then the PATH is searched. When a reg\_exe is given the fileid cannot contain a path.

## Output a Default String

25 Default simply outputs text to the output file if none of the previous options were successful.

Syntax:  
;default;output

### Output

This is the output that will be placed in the output  
5 file. There are several variables that will be substituted  
into this string before it is returned.

%appid%

The current application id.

10 %del%

The input field delimiter.

To summarize, there are several search options which  
could be employed to determine a version or other state  
information for a computer system, including:

- 15
- Searching for a file
  - Searching for a certain INI file entry
  - Using output of some other external application
  - Checking for a default condition, or
  - Looking for a registry key/value

20

FIGS. 3A & 3B depict a flowchart of one embodiment for  
processing a dictionary file data structure in accordance  
with the principles of the present invention. The data  
structure in this example employs the first four options  
25 noted above. Other examples, however, will be apparent to  
those skilled in the art.

In the embodiment of FIGS. 3A & 3B, dictionary file  
processing 30 begins with setting of a "Looking For a New

Group" flag 40. As noted above, the dictionary file can be organized into multiple subject groups and the new group flag is set to indicate that the processing is searching for a next group in the file. Next, processing inquiries whether  
5 there are any records or lines in the dictionary file left to be processed 50. If "no", then the results obtained up to this point are output in a file to the information repository 60 and processing is terminated 70.

Again, as used herein, a "record" is a line of inquiry  
10 which may comprise an "instruction" within the dictionary file. An instruction is a line of inquiry which contains a condition that when satisfied causes a result to be output for inclusion as a line in the file comprising the state information of the computer system under examination. In one  
15 example, each record or line of inquiry also comprises an instruction. If records remain, then processing reads a next record from the file 80 and determines whether the "Looking For New Group" flag is set 90. If so, then the new record may comprise the first line of the new group and inquiry  
20 determines whether the new group is found 100. If "no", then processing returns to determine whether any records remain in the dictionary file 50. If a new group has been found, then the group substitution variables for the new group are set 110. The group substitution variables could comprise values  
25 for the group which are to be inserted into the output with the state information. The "Looking For New Group" flag is then reset 120 and processing determines whether there are any records remaining 50. If so, then at the next pass through inquiry 90 the "Looking For New Group" flag will not



be set and processing determines whether a new group is found 130. If a new group is found, this means that the previous group had no lines of inquiry and the group substitution variables are set for the new group 110.

5        Assuming that the record does not comprise a new group, then processing determines whether the record is a file check inquiry 140. If so, then a process file inquiry subroutine 150 is called. If the record is not a file check inquiry, processing determines whether the record is an INI file  
10    content check inquiry 160. If so, then a process INI inquiry subroutine 170 is called. If the answer to inquiry 160 is "no", then processing determines whether the record is an external process check inquiry 180, and if so, a process external process inquiry subroutine 190 is called. Finally,  
15    in this example, processing determines whether the record is a default inquiry 200. If so, the process default inquiry subroutine 210 is called. Otherwise, no action is taken and processing determines whether there are any records remaining 50.

20        FIGS. 4, 5, 6 & 7 respectively depict one embodiment of a process file inquiry subroutine 150, a process INI inquiry subroutine 170, a process external process inquiry subroutine 190, and a process default inquiry subroutine 210. Continuing with FIGS. 3A & 3B, after processing the record  
25    through the appropriate subroutine, inquiry is made whether the condition within the record has been satisfied 220. If so, then the result associated with the record is added to a results file to be output upon completion of dictionary file

processing 230. The "Look For New Group" flag is then set 240 and processing looks for a next record in the dictionary file 50. If the condition in the record is unsuccessful, then processing directly returns to inquiry 50.

5        FIG. 4 depicts one embodiment of a process file inquiry subroutine 400 which begins by checking whether there are more files in the inquiry to check 410. If "no", then the result or variables from the inquiry line are substituted into the result line to be output as the state information  
10 (from the processing of FIGS. 3A & 3B). The result line would also include the group substitution variables set during the processing of FIGS. 3A & 3B. This successfully ends processing 425 of the process file inquiry 400. If there are more files to be checked in the inquiry, processing  
15 determines whether the desired file exists 430, whether the size of the file matches the anticipated size 440, whether the date of the file matches 450 and whether the time of the file matches 460. If "yes" to these inquiries, then the condition has been satisfied and processing determines  
20 whether there are any more files to check. If "no", then the corresponding result of the inquiry is substituted into the result line to be output as the state information for the client and processing successfully terminates 425. If the answer to any of inquiry 430, 440, 450 or 460 is "no", then  
25 processing is terminated for the line of inquiry since the condition does not exist on the client 475.

FIG. 5 depicts one embodiment of a process INI inquiry routine 500 which begins by opening the INI file 510. Processing determines whether there are more lines to check  
30 in the INI file 520. If "no", then the process INI inquiry was not successful 525. Otherwise, the next line of the INI

file is read 530 and processing determines whether the line  
is the start of a desired application or section in the INI  
file 540. If "no", then a next line in the INI file is  
checked and the process is repeated until the start of the  
5 desired application in the INI file is found.

Once the start of the desired application is found,  
processing determines whether there are more lines in the INI  
file to check 550. If not, then the process INI inquiry has  
been unsuccessful 525. If so, the next line is read 560 and  
10 processing determines whether the variable matches 570. If  
"no", then a next line is checked 550. Otherwise, processing  
inquires whether the value of the variable matches or is a  
don't care 580. If "no", then a next line is checked. If  
the value matches or is a don't care, then the variables are  
15 substituted into the result line to be output as a portion of  
the state information 590 and the process INI inquiry is  
successfully ended 595.

FIG. 6 depicts one embodiment of a process external  
process inquiry 600 which calls the external process and  
20 places the result in a temporary file 610. As noted above,  
there are a number of options with respect to processing of  
an external application. The external application could  
output an INI file, an ASCII file, or registry changes. In  
this example, an INI file output is assumed. Thus, an INI  
25 inquiry is processed on the temporary file 620 to determine  
whether the condition has been successfully met. The  
temporary file is then deleted 630 and processing determines  
whether the INI inquiry was successful 640. If so, an end  
successful result 655 is obtained, otherwise an end not  
30 successful result is output 645.

FIG. 7 is a flowchart for processing a default inquiry 700. This processing simply comprises substituting variables into the result line 710 as defined by the default inquiry. Processing is always successful 715 with a default inquiry.

5        FIG. 8 depicts an example dictionary file wherein an external process IPCconfig is called and the results of that external process are compared with a predefined condition. This example dictionary file is described in greater detail in the above-incorporated, co-filed patent application.

10        Those skilled in the art will note from the above description that presented herein is an information gathering facility which employs a dictionary file data structure at a central location to derive state information on one or more computer systems in a network. Advantageously, the format of  
15        the state information returned can be defined by results contained within the dictionary file data structure itself. The data structure comprises a plurality of inquiries which are organized into subject groups and when a particular condition of an inquiry is satisfied, all other inquiries in  
20        the group are skipped. As new versions of an application on one or more computer systems in the network become available, a new inquiry is simply added to the respective group that is looking for that application.

25        The present invention can be included, for example, in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. This media has embodied therein, for instance, computer readable program code means for providing and facilitating the

capabilities of the present invention. The articles of manufacture can be included as part of the computer system or sold separately.

Additionally, at least one program storage device  
5 readable by machine, tangibly embodying at least one program of instructions executable by the machine, to perform the capabilities of the present invention, can be provided.

The flow diagrams depicted herein are provided by way of example. There may be variations to these diagrams or the  
10 steps (or operations) described herein without departing from the spirit of the invention. For instance, in certain cases, the steps may be performed in differing order, or steps may be added, deleted or modified. All of these variations are considered to comprise part of the present invention as  
15 recited in the appended claims.

While the invention has been described in detail herein in accordance with certain preferred embodiments thereof, many modifications and changes therein may be effected by those skilled in the art. Accordingly, it is intended by the  
20 appended claims to cover all such modifications and changes as fall within the true spirit and scope of the invention.

## Claims

1           1.    A method for gathering information on a state of a  
2 computer system, said method comprising:

3                providing a dictionary file having a plurality of  
4 inquiries for ascertaining state information on said  
5 computer system, said plurality of inquiries being  
6 organized into at least one subject group, each subject  
7 group being directed to a different piece of said state  
8 information, at least one group of said at least one  
9 subject group having multiple records of inquiry; and

10              processing at least one inquiry of said plurality  
11 of inquiries of said dictionary file to accumulate said  
12 state information, said processing comprising for each  
13 group of said at least one group having multiple records  
14 of inquiry:

15                   processing a record of said multiple records  
16 of inquiry, and if a condition of said record is  
17 satisfied then terminating processing of said  
18 group, otherwise processing a next record of said  
19 multiple records of inquiry and continuing until a  
20 condition of one record of said multiple records of  
21 inquiry is satisfied or all records of said  
22 multiple records of inquiry of said group have been  
23 processed.

1           2.    The method of claim 1, wherein said at least one  
2 subject group comprises multiple subject groups, and wherein  
3 said processing of each group of said at least one group  
4 having multiple records of inquiry comprises proceeding to a  
5 next group of said multiple subject groups when said  
6 condition of one record of said multiple records of inquiry  
7 in said at least one group is satisfied.

1           3.    The method of claim 1, wherein at least some  
2 inquiries of said plurality of inquiries comprise  
3 instructions, each instruction providing a result when a  
4 condition of said instruction is satisfied.

1           4.    The method of claim 3, further comprising  
2 collecting results of said instructions into a file, said  
3 file being representative of said state of said computer  
4 system.

1           5.    The method of claim 4, wherein each inquiry of said  
2 plurality of inquiries is an instruction which provides a  
3 result when a condition of said instruction is satisfied.

1           6.    The method of claim 4, wherein said processing  
2 comprises processing each group of said at least one subject  
3 group, and wherein said method further comprises transferring  
4 said file to an information repository coupled to said  
5 computer system across a network.

1           7.    The method of claim 6, wherein said computer system  
2 comprises one computer system of a plurality of computer  
3 systems coupled to said network.

1        8.    The method of claim 1, wherein said dictionary file  
2 comprises a rules database in an ASCII file.

1        9.    The method of claim 1, wherein at least one record  
2 of inquiry of said multiple records of inquiry comprises an  
3 instruction which provides a result when a condition of said  
4 instruction is satisfied, said result comprising state  
5 information for said group having said record when the  
6 condition of said instruction is satisfied.

1        10.   The method of claim 1, wherein said computer system  
2 comprises one computer system within a network of computer  
3 systems, and wherein said providing comprises reading said  
4 dictionary file from a server coupled to said network of  
5 computer systems to said one computer system to gather said  
6 state information thereon.

1        11.   The method of claim 10, further comprising  
2 forwarding results representative of gathered state  
3 information to an information repository coupled to said  
4 network of computer systems, said information repository  
5 residing at said server system providing said dictionary  
6 file.

1        12.   The method of claim 1, wherein said processing  
2 comprises processing each group of said at least one subject  
3 group, and setting group substitution variables for output  
4 upon initiation of processing of each group of said at least  
5 one subject group.



1           13. The method of claim 1, wherein said multiple  
2 records of inquiry of said at least one group comprise at  
3 least one of a file check inquiry, a file content check  
4 inquiry, an external process check inquiry, or a default  
5 inquiry for said group.

13. The method of claim 1, wherein said multiple records of inquiry of said at least one group comprise at least one of a file check inquiry, a file content check inquiry, an external process check inquiry, or a default inquiry for said group.

1        14. The method of claim 1, wherein said plurality of  
2 inquiries comprise multiple inquiry types, and wherein said  
3 multiple inquiry types comprise at least some of:

4            a file inquiry which checks for existence of a file  
5 of a certain date, time or size and which can return  
6 file information;

7            an INI file inquiry which checks for a certain  
8 application, variable and value, and which can return a  
9 certain value or one or more variables and values;

10           an ASCII file inquiry which checks for a certain  
11 character string in a file, and which can return  
12 information on a line within the character string;

13           a registry inquiry which checks for a certain  
14 registry tree or value, and which can return one or more  
15 values in a tree or sub-tree;

16           an external process inquiry using an INI output,  
17 which comprises executing an external process and  
18 performing an INI file inquiry on the result;

19           an external process inquiry which executes an  
20 external process, provides an ASCII output, and performs  
21 an ASCII file inquiry on the result;

22           an external process inquiry using a registry, which  
23 executes an external process and performs a registry  
24 inquiry on the result; and

25           multiple inquiries which comprise a combination of  
26 multiple other inquiry types, where all must succeed.

1        15. A method for gathering information on a state of a  
2 computer system, said method comprising:

3            providing a dictionary file having a plurality of  
4 inquiries for ascertaining said state information on  
5 said computer system, at least one inquiry of said  
6 plurality of inquiries comprising an instruction having  
7 a result which is output when a condition of said  
8 instruction is satisfied; and

9            processing at least one inquiry of said plurality  
10 of inquiries of said dictionary file to accumulate said  
11 state information, said processing comprising for each  
12 instruction, outputting said result of said instruction  
13 from said dictionary file when said condition of said  
14 instruction is satisfied, wherein said state information  
15 on said computer system comprises outputted results from  
16 satisfaction of said at least one instruction.

1        16. The method of claim 15, wherein said at least one  
2 inquiry comprising said instruction comprises multiple  
3 inquiries of said plurality of inquiries, each instruction  
4 having a result which is output when a condition of said  
5 instruction is satisfied.

1        17. The method of claim 16, further comprising  
2 collecting results of said instructions into a file, said  
3 file being representative of said state of the computer  
4 system.

1        18. The method of claim 17, wherein each inquiry of  
2 said plurality of inquiries is an instruction which provides  
3 a result when a condition of said instruction is satisfied.

1        19. The method of claim 17, further comprising  
2 transferring said file to an information repository coupled  
3 to said computer system across a network.

1        20. The method of claim 19, wherein said computer  
2 system comprises one computer system of a plurality of  
3 computer systems coupled to said network.

1        21. The method of claim 15, wherein said dictionary  
2 file comprises a rules database defined in an ASCII file.

1        22. The method of claim 15, wherein said computer  
2 system comprises one computer system within a network of  
3 computer systems, and wherein said providing comprises  
4 reading said dictionary file from a server coupled to said  
5 network to said one computer system to gather said state  
6 information thereon.

1        23. The method of claim 22, wherein said reading of  
2 said dictionary file is performed by an inquiry tool routine  
3 located on said one computer system.

1        24. The method of claim 22, further comprising  
2 forwarding results representative of gathered state  
3 information to an information repository of said network of  
4 computer systems, said information repository residing at  
5 said server providing said dictionary file.

1        25. The method of claim 15, wherein said plurality of  
2 inquiries comprise at least one of a file check inquiry, a  
3 file content check inquiry, an external process check  
4 inquiry, or a default inquiry.

1        26. The method of claim 15, wherein said plurality of  
2 inquiries comprise multiple inquiry types, and wherein said  
3 multiple inquiry types comprise at least some of:

4            a file inquiry which checks for existence of a file  
5 of a certain date, time or size and which can return  
6 file information;

7            an INI file inquiry which checks for a certain  
8 application, variable and value, and which can return a  
9 certain value, or one or more variables and values;

10           an ASCII file inquiry which checks for a certain  
11 character string in a file, and which can return  
12 information on a line within the character string;

13           a registry inquiry which checks for a certain  
14 registry tree or value, and which can return one or more  
15 values in a tree or sub-tree;

16           an external process inquiry using an INI output,  
17 which comprises executing an external process and  
18 performing an INI file inquiry on the result;

19           an external process inquiry which executes an  
20 external process, provides an ASCII output, and performs  
21 an ASCII file inquiry on the result;

22           an external process inquiry using a registry, which  
23 executes an external process and performs a registry  
24 inquiry on the result; and

25           multiple inquiries which comprise a combination of  
26 multiple other inquiry types, where all must succeed.

1        27. The method of claim 15, wherein said plurality of  
2 inquiries are organized into at least one subject group, each  
3 subject group being directed to a different piece of said  
4 state information, at least one group of the at least one  
5 subject group having multiple instructions, and processing  
6 each instruction of each group of the at least one group  
7 having multiple instructions such that if a condition of the  
8 instruction is satisfied then terminating processing of the  
9 group, otherwise processing a next instruction of the  
10 multiple instructions within the group and continuing until a  
11 condition of one instruction of the multiple instructions is  
12 satisfied or until all instructions of the multiple  
13 instructions of the group have been processed.

1        28. A memory for storing a dictionary file data  
2 structure, the dictionary file data structure facilitating  
3 gathering of information on a state of a computer system, the  
4 dictionary file data structure comprising:

5            a plurality of inquiries for ascertaining state  
6 information on the computer system, said plurality of  
7 inquiries being organized into at least one subject  
8 group, each subject group being directed to a different  
9 piece of said state information, at least one group of  
10 the at least one subject group having multiple records  
11 of inquiry; and

12            wherein the multiple records of inquiry of the at  
13 least one group comprise multiple instructions, each  
14 instruction comprising a result to be output when a  
15 condition of said instruction is satisfied, wherein  
16 outputting of a result from one instruction terminates  
17 processing of said at least one group having multiple  
18 records of inquiry.

1        29. A memory for storing a dictionary file data  
2 structure, the dictionary file data structure facilitating  
3 gathering information on a state of a computer system, the  
4 dictionary file data structure comprising:

5            a plurality of inquiries for ascertaining said  
6 state information on said computer system, at least one  
7 inquiry of the plurality of inquiries comprising an  
8 instruction having a result which is output when a  
9 condition of said instruction is satisfied.

1        30. A system for gathering information on a state of a  
2 computer system, said system comprising:

3            means for providing a dictionary file having a  
4 plurality of inquiries for ascertaining state  
5 information on said computer system, said plurality of  
6 inquiries being organized into at least one subject  
7 group, each subject group being directed to a different  
8 piece of said state information, at least one group of  
9 said at least one subject group having multiple records  
10 of inquiry; and

11           means for processing at least one inquiry of said  
12 plurality of inquiries of said dictionary file to  
13 accumulate said state information, said means for  
14 processing comprises for each group of said at least one  
15 group having multiple records of inquiry:

16            means for processing a record of said multiple  
17 records of inquiry, and if a condition of said  
18 record is satisfied then for terminating processing  
19 of said group, otherwise for processing a next  
20 record of said multiple records of inquiry and  
21 continuing until a condition of one record of said  
22 multiple records of inquiry is satisfied or all  
23 records of said multiple records of inquiry of said  
24 group have been processed.



1        31. The system of claim 30, wherein said at least one  
2 subject group comprises multiple subject groups, and wherein  
3 said means for processing of each group of said at least one  
4 group having multiple records of inquiry comprises means for  
5 proceeding to a next group of said multiple subject groups  
6 when said condition of one record of said multiple records of  
7 inquiry in said at least one group is satisfied.

1        32. The system of claim 30, wherein at least some  
2 inquiries of said plurality of inquiries comprise  
3 instructions, each instruction providing a result when a  
4 condition of said instruction is satisfied.

1        33. The system of claim 32, further comprising means  
2 for collecting results of said instructions into a file, said  
3 file being representative of said state of said computer  
4 system.

1        34. The system of claim 33, wherein each inquiry of  
2 said plurality of inquiries is an instruction which provides  
3 a result when a condition of said instruction is satisfied.

1        35. The system of claim 33, wherein said means for  
2 processing comprises means for processing each group of said  
3 at least one subject group, and wherein said system further  
4 comprises means for transferring said file to an information  
5 repository coupled to said computer system across a network.

1        36. The system of claim 35, wherein said computer  
2 system comprises one computer system of a plurality of  
3 computer systems coupled to said network.

1        37. The system of claim 30, wherein said dictionary  
2 file comprises a rules database in an ASCII file.

1        38. The system of claim 30, wherein said means for  
2 processing comprises means for processing each group of said  
3 at least one subject group, and means for setting group  
4 substitution variables for output upon initiation of  
5 processing of each group of said at least one subject group.

1        39. The system of claim 30, wherein said multiple  
2 records of inquiry of said at least one group comprise at  
3 least one of a file check inquiry, a file content check  
4 inquiry, an external process check inquiry, or a default  
5 inquiry for said group.

1        40. The system of claim 30, wherein said plurality of  
2 inquiries comprise multiple inquiry types, and wherein said  
3 multiple inquiry types comprise at least some of:

4                a file inquiry which checks for existence of a file  
5 of a certain date, time or size and which can return  
6 file information;

7                an INI file inquiry which checks for a certain  
8 application, variable and value, and which can return a  
9 certain value or one or more variables and values;

10               an ASCII file inquiry which checks for a certain  
11 character string in a file, and which can return  
12 information on a line within the character string;

13           a registry inquiry which checks for a certain  
14 registry tree or value, and which can return one or more  
15 values in a tree or sub-tree;

16           an external process inquiry using an INI output,  
17 which comprises executing an external process and  
18 performing an INI file inquiry on the result;

19           an external process inquiry which executes an  
20 external process, provides an ASCII output, and performs  
21 an ASCII file inquiry on the result;

22           an external process inquiry using a registry, which  
23 executes an external process and performs a registry  
24 inquiry on the result; and

25           multiple inquiries which comprise a combination of  
26 multiple other inquiry types, where all must succeed.

1           41. A system for gathering information on a state of a  
2 computer system, said system comprising:

3           means for providing a dictionary file having a  
4 plurality of inquiries for ascertaining said state  
5 information on said computer system, at least one  
6 inquiry of said plurality of inquiries comprising an  
7 instruction having a result which is output when a  
8 condition of said instruction is satisfied; and

9           means for processing said at least one inquiry of  
10 said plurality of inquiries of said dictionary file to  
11 accumulate said state information, said means for  
12 processing comprising for each instruction, means for  
13 outputting said result of said instruction from said  
14 dictionary file when said condition of said instruction  
15 is satisfied, wherein said state information on said  
16 computer system comprises any outputted results from  
17 satisfaction of said at least one instruction.

1           42. The system of claim 41, wherein said at least one  
2 inquiry comprising said instruction comprises multiple  
3 inquiries of said plurality of inquiries, each instruction  
4 having a result which is output when a condition of said  
5 instruction is satisfied.

1           43. The system of claim 42, further comprising means  
2 for collecting results of said instructions into a file, said  
3 file being representative of said state of the computer  
4 system.

1        44. The system of claim 43, wherein each inquiry of  
2 said plurality of inquiries is an instruction which provides  
3 a result when a condition of said instruction is satisfied.

1        45. The system of claim 43, further comprising means  
2 for transferring said file to an information repository  
3 coupled to said computer system across a network.

1        46. The system of claim 45, wherein said computer  
2 system comprises one computer system of a plurality of  
3 computer systems coupled to said network.

1        47. The system of claim 41, wherein said dictionary  
2 file comprises a rules database defined in an ASCII file.

1        48. The system of claim 41, wherein said plurality of  
2 inquiries comprise at least one of a file check inquiry, a  
3 file content check inquiry, an external process check  
4 inquiry, or a default inquiry.

1        49. The system of claim 41, wherein said plurality of  
2 inquiries comprise multiple inquiry types, and wherein said  
3 multiple inquiry types comprise at least some of:

4            a file inquiry which checks for existence of a file  
5 of a certain date, time or size and which can return  
6 file information;

7            an INI file inquiry which checks for a certain  
8 application, variable and value, and which can return a  
9 certain value, or one or more variables and values;

10           an ASCII file inquiry which checks for a certain  
11           character string in a file, and which can return  
12           information on a line within the character string;

13           a registry inquiry which checks for a certain  
14           registry tree or value, and which can return one or more  
15           values in a tree or sub-tree;

16           an external process inquiry using an INI output,  
17           which comprises executing an external process and  
18           performing an INI file inquiry on the result;

19           an external process inquiry which executes an  
20           external process, provides an ASCII output, and performs  
21           an ASCII file inquiry on the result;

22           an external process inquiry using a registry, which  
23           executes an external process and performs a registry  
24           inquiry on the result; and

25           multiple inquiries which comprise a combination of  
26           multiple other inquiry types, where all must succeed.

1        50. The system of claim 41, wherein said plurality of  
2 inquiries are organized into at least one subject group, each  
3 subject group being directed to a different piece of said  
4 state information, at least one group of the at least one  
5 subject group having multiple instructions, and said system  
6 comprising means for processing each instruction of each  
7 group of the at least one group having multiple instructions  
8 such that if a condition of the instruction is satisfied  
9 processing of the group is terminated, otherwise a next  
10 instruction of the multiple instructions within the group is  
11 processed and continuing until a condition of one instruction  
12 of the multiple instructions is satisfied or until all  
13 instructions of the multiple instructions of the group have  
14 been processed.

1        51. At least one program storage device readable by a  
2 machine, tangibly embodying at least one program of  
3 instructions executable by the machine to perform a method  
4 for gathering information on a state of a computer system,  
5 said method comprising:

6            providing a dictionary file having a plurality of  
7 inquiries for ascertaining state information on said  
8 computer system, said plurality of inquiries being  
9 organized into at least one subject group, each subject  
10 group being directed to a different piece of said state  
11 information, at least one group of said at least one  
12 subject group having multiple records of inquiry; and

13           processing at least one inquiry of said plurality  
14 of inquiries of said dictionary file to accumulate said  
15 state information, said processing comprising for each  
16 group of said at least one group having multiple records  
17 of inquiry:

18                processing a record of said multiple records  
19 of inquiry, and if a condition of said record is  
20 satisfied then terminating processing of said  
21 group, otherwise processing a next record of said  
22 multiple records of inquiry and continuing until a  
23 condition of one record of said multiple records of  
24 inquiry is satisfied or all records of said  
25 multiple records of inquiry of said group have been  
26 processed.



1        52. The at least one program storage device of claim  
2 51, wherein said at least one subject group comprises  
3 multiple subject groups, and wherein said processing of each  
4 group of said at least one group having multiple records of  
5 inquiry comprises proceeding to a next group of said multiple  
6 subject groups when said condition of one record of said  
7 multiple records of inquiry in said at least one group is  
8 satisfied.

1        53. The at least one program storage device of claim  
2 51, wherein at least some inquiries of said plurality of  
3 inquiries comprise instructions, each instruction providing a  
4 result when a condition of said instruction is satisfied.

1        54. The at least one program storage device of claim  
2 53, further comprising collecting results of said  
3 instructions into a file, said file being representative of  
4 said state of said computer system.

1        55. The at least one program storage device of claim  
2 54, wherein each inquiry of said plurality of inquiries is an  
3 instruction which provides a result when a condition of said  
4 instruction is satisfied.

1        56. The at least one program storage device of claim  
2 54, wherein said processing comprises processing each group  
3 of said at least one subject group, and wherein said method  
4 further comprises transferring said file to an information  
5 repository coupled to said computer system across a network.

1        57. The at least one program storage device of claim  
2 56, wherein said computer system comprises one computer  
3 system of a plurality of computer systems coupled to said  
4 network.

1        58. The at least one program storage device of claim  
2 51, wherein said dictionary file comprises a rules database  
3 in an ASCII file.

1        59. The at least one program storage device of claim  
2 51, wherein said multiple records of inquiry of said at least  
3 one group comprise at least one of a file check inquiry, a  
4 file content check inquiry, an external process check  
5 inquiry, or a default inquiry for said group.

1        60. The at least one program storage device of claim  
2 51, wherein said plurality of inquiries comprise multiple  
3 inquiry types, and wherein said multiple inquiry types  
4 comprise at least some of:

5            a file inquiry which checks for existence of a file  
6 of a certain date, time or size and which can return  
7 file information;

8            an INI file inquiry which checks for a certain  
9 application, variable and value, and which can return a  
10 certain value or one or more variables and values;

11           an ASCII file inquiry which checks for a certain  
12 character string in a file, and which can return  
13 information on a line within the character string;

14           a registry inquiry which checks for a certain  
15 registry tree or value, and which can return one or more  
16 values in a tree or sub-tree;

17           an external process inquiry using an INI output,  
18 which comprises executing an external process and  
19 performing an INI file inquiry on the result;

20           an external process inquiry which executes an  
21 external process, provides an ASCII output, and performs  
22 an ASCII file inquiry on the result;

23           an external process inquiry using a registry, which  
24 executes an external process and performs a registry  
25 inquiry on the result; and

26           multiple inquiries which comprise a combination of  
27 multiple other inquiry types, where all must succeed.

1        61. At least one program storage device readable by a  
2 machine, tangibly embodying at least one program of  
3 instructions executable by the machine to perform a method  
4 for gathering information on a state of a computer system,  
5 said method comprising:

6            providing a dictionary file having a plurality of  
7 inquiries for ascertaining said state information on  
8 said computer system, at least one inquiry of said  
9 plurality of inquiries comprising an instruction having  
10 a result which is output when a condition of said  
11 instruction is satisfied; and

12           processing at least one inquiry of said plurality  
13 of inquiries of said dictionary file to accumulate said  
14 state information, said processing comprising for each  
15 instruction, outputting said result of said instruction  
16 from said dictionary file when said condition of said  
17 instruction is satisfied, wherein said state information  
18 on said computer system comprises outputted results from  
19 satisfaction of said at least one instruction.

1        62. The at least one program storage device of claim  
2 61, wherein said at least one inquiry comprising said  
3 instruction comprises multiple inquiries of said plurality of  
4 inquiries, each instruction having a result which is output  
5 when a condition of said instruction is satisfied.

1        63. The at least one program storage device of claim  
2 62, further comprising collecting results of said  
3 instructions into a file, said file being representative of  
4 said state of the computer system.

1        64. The at least one program storage device of claim  
2 63, wherein each inquiry of said plurality of inquiries is an  
3 instruction which provides a result when a condition of said  
4 instruction is satisfied.

1        65. The at least one program storage device of claim  
2 63, further comprising transferring said file to an  
3 information repository coupled to said computer system across  
4 a network.

1        66. The at least one program storage device of claim  
2 65, wherein said computer system comprises one computer  
3 system of a plurality of computer systems coupled to said  
4 network.

1        67. The at least one program storage device of claim  
2 61, wherein said dictionary file comprises a rules database  
3 defined in an ASCII file.

1        68. The at least one program storage device of claim  
2 61, wherein said plurality of inquiries comprise at least one  
3 of a file check inquiry, a file content check inquiry, an  
4 external process check inquiry, or a default inquiry.

1       69. The at least one program storage device of claim  
2 61, wherein said plurality of inquiries comprise multiple  
3 inquiry types, and wherein said multiple inquiry types  
4 comprise at least some of:

5           a file inquiry which checks for existence of a file  
6 of a certain date, time or size and which can return  
7 file information;

8           an INI file inquiry which checks for a certain  
9 application, variable and value, and which can return a  
10 certain value, or one or more variables and values;

11          an ASCII file inquiry which checks for a certain  
12 character string in a file, and which can return  
13 information on a line within the character string;

14          a registry inquiry which checks for a certain  
15 registry tree or value, and which can return one or more  
16 values in a tree or sub-tree;

17          an external process inquiry using an INI output,  
18 which comprises executing an external process and  
19 performing an INI file inquiry on the result;

20          an external process inquiry which executes an  
21 external process, provides an ASCII output, and performs  
22 an ASCII file inquiry on the result;

23          an external process inquiry using a registry, which  
24 executes an external process and performs a registry  
25 inquiry on the result; and

26 multiple inquiries which comprise a combination of  
27 multiple other inquiry types, where all must succeed.

1 70. The at least one program storage device of claim  
2 61, wherein said plurality of inquiries are organized into at  
3 least one subject group, each subject group being directed to  
4 a different piece of said state information, at least one  
5 group of the at least one subject group having multiple  
6 instructions, and processing each instruction of each group  
7 of the at least one group having multiple instructions such  
8 that if a condition of the instruction is satisfied then  
9 terminating processing of the group, otherwise processing a  
10 next instruction of the multiple instructions within the  
11 group and continuing until a condition of one instruction of  
12 the multiple instructions is satisfied or until all  
13 instructions of the multiple instructions of the group have  
14 been processed.

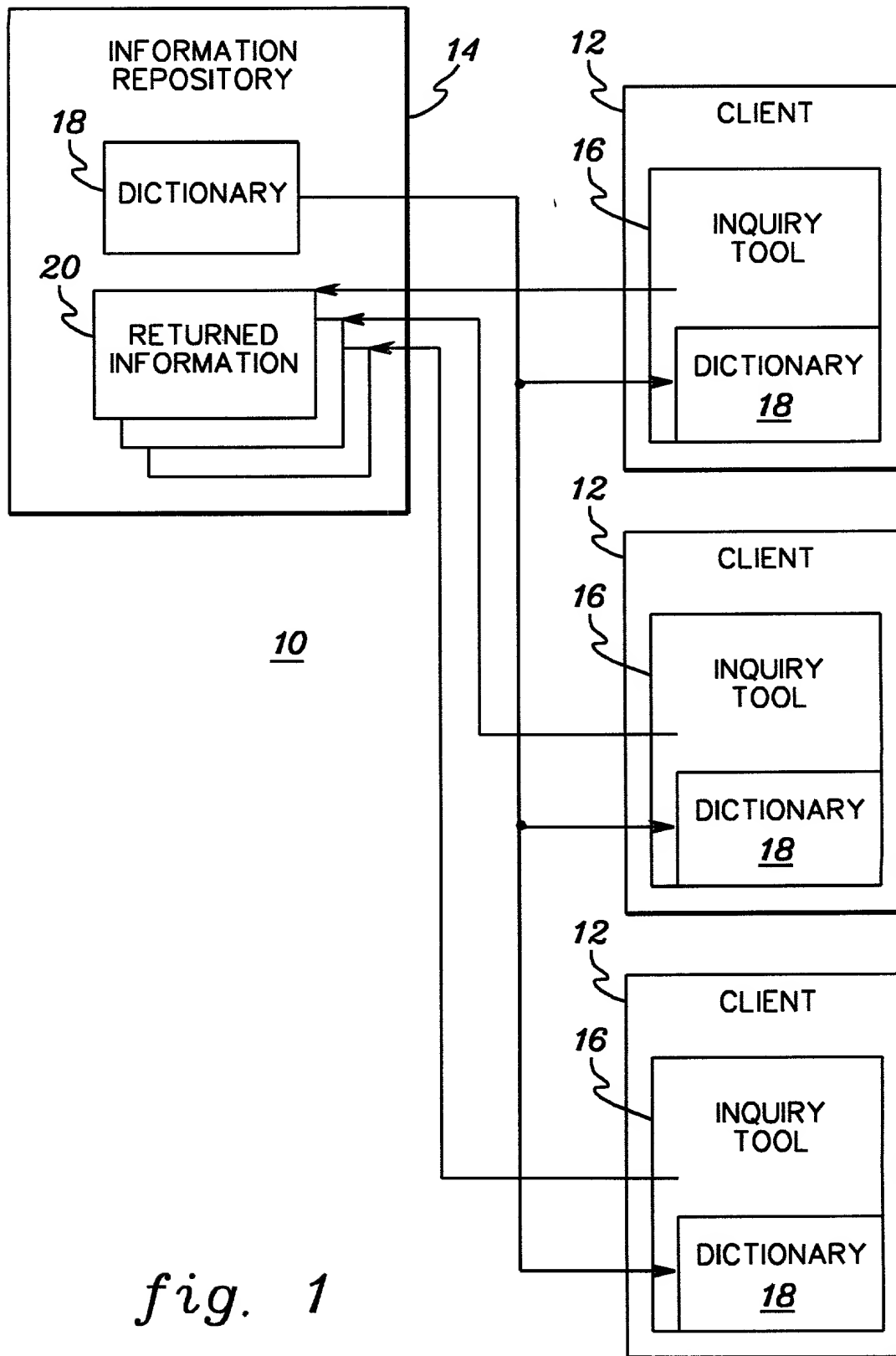
\* \* \* \* \*

**INFORMATION GATHERING FACILITY EMPLOYING  
DICTIONARY FILE CONTAINING MULTIPLE INQUIRIES**

**Abstract of the Invention**

An information gathering facility is provided to gather  
5 information on the state of a computer system. The  
information gathering facility includes a dictionary file  
data structure having a plurality of inquiries for  
ascertaining state information on the system.  
The plurality of inquiries are organized into at least one  
10 subject group. Each subject group is directed to a different  
piece of the state information. At least one group has  
multiple records of inquiry. These multiple records of  
inquiry are processed such that if a condition of one record  
is satisfied then processing terminates for the group,  
15 otherwise processing proceeds to a next record and continues  
until a condition of one of the records of the group is  
satisfied or until all records have been processed. One or  
more of the plurality of inquiries comprise instructions  
which output a result when a condition of the instruction is  
20 satisfied. Thus, the state information to be output is  
defined by the results contained within the instructions of  
the dictionary file itself.



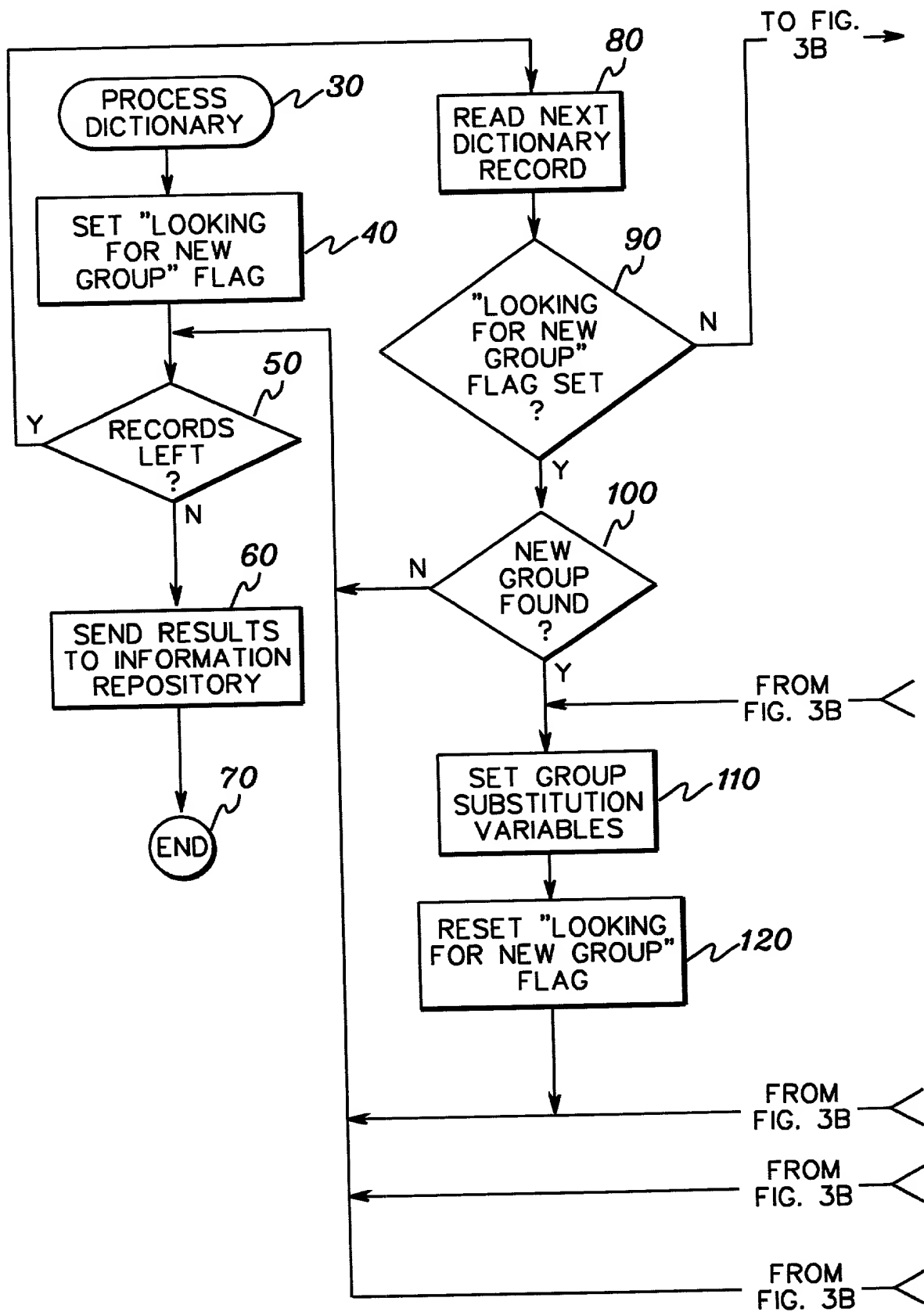


*fig. 1*

## POSSIBLE INQUIRY TYPES

- FILE
  - ✓ CHECK FOR EXISTENCE OF A FILE OF A CERTAIN DATE, TIME AND SIZE.
  - ✓ CAN RETURN FILE INFORMATION.
- INI FILE
  - ✓ CHECK FOR A CERTAIN APPLICATION, VARIABLE AND VALUE.
  - ✓ CAN RETURN A CERTAIN VALUE OR ALL VARIABLES AND VALUES.
- ASCII FILE
  - ✓ CHECK FOR A CERTAIN CHARACTER STRING IN A FILE.
  - ✓ CAN RETURN INFORMATION ON THE LINE WITH THE CHARACTER STRING.
- REGISTRY
  - ✓ CHECK FOR A CERTAIN REGISTRY TREE AND VALUE.
  - ✓ CAN RETURN VALUE OR ALL VALUES IN SUB TREES.
- EXTERNAL PROCESS USING INI OUTPUT
  - ✓ EXECUTE AN EXTERNAL PROCESS AND DO AN INI FILE INQUIRY ON THE RESULTS.
- EXTERNAL PROCESS USING ASCII OUTPUT
  - ✓ EXECUTE AN EXTERNAL PROCESS AND DO AN ASCII FILE INQUIRY ON THE RESULTS.
- EXTERNAL PROCESS USING REGISTRY
  - ✓ EXECUTE AN EXTERNAL PROCESS AND DO A REGISTRY INQUIRY ON THE RESULTS.
- MULTIPLE
  - ✓ DO A COMBINATION OF THE OTHER INQUIRES WHERE ALL MUST SUCCEED.

*fig. 2*



*fig. 3A*

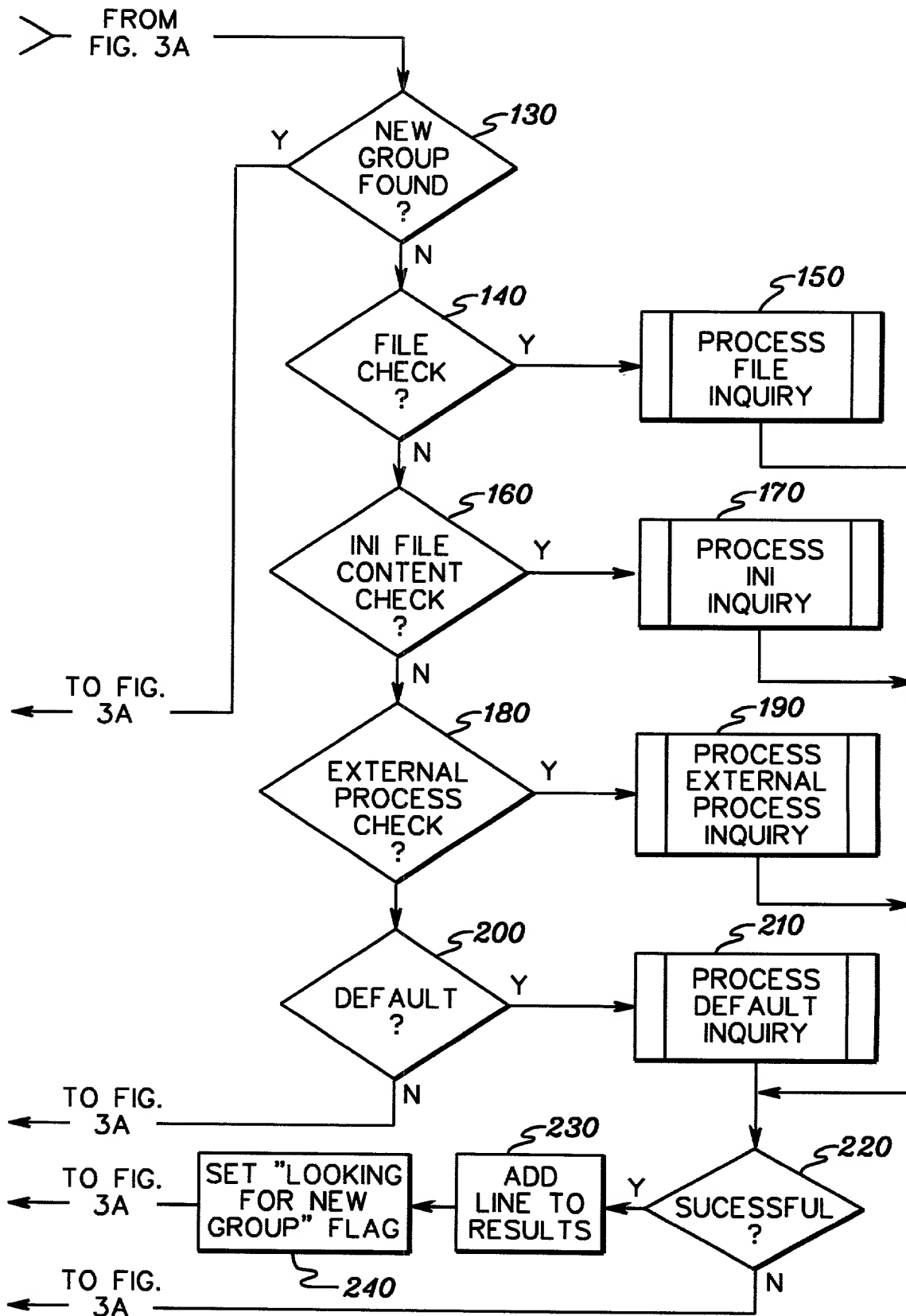


fig. 3B

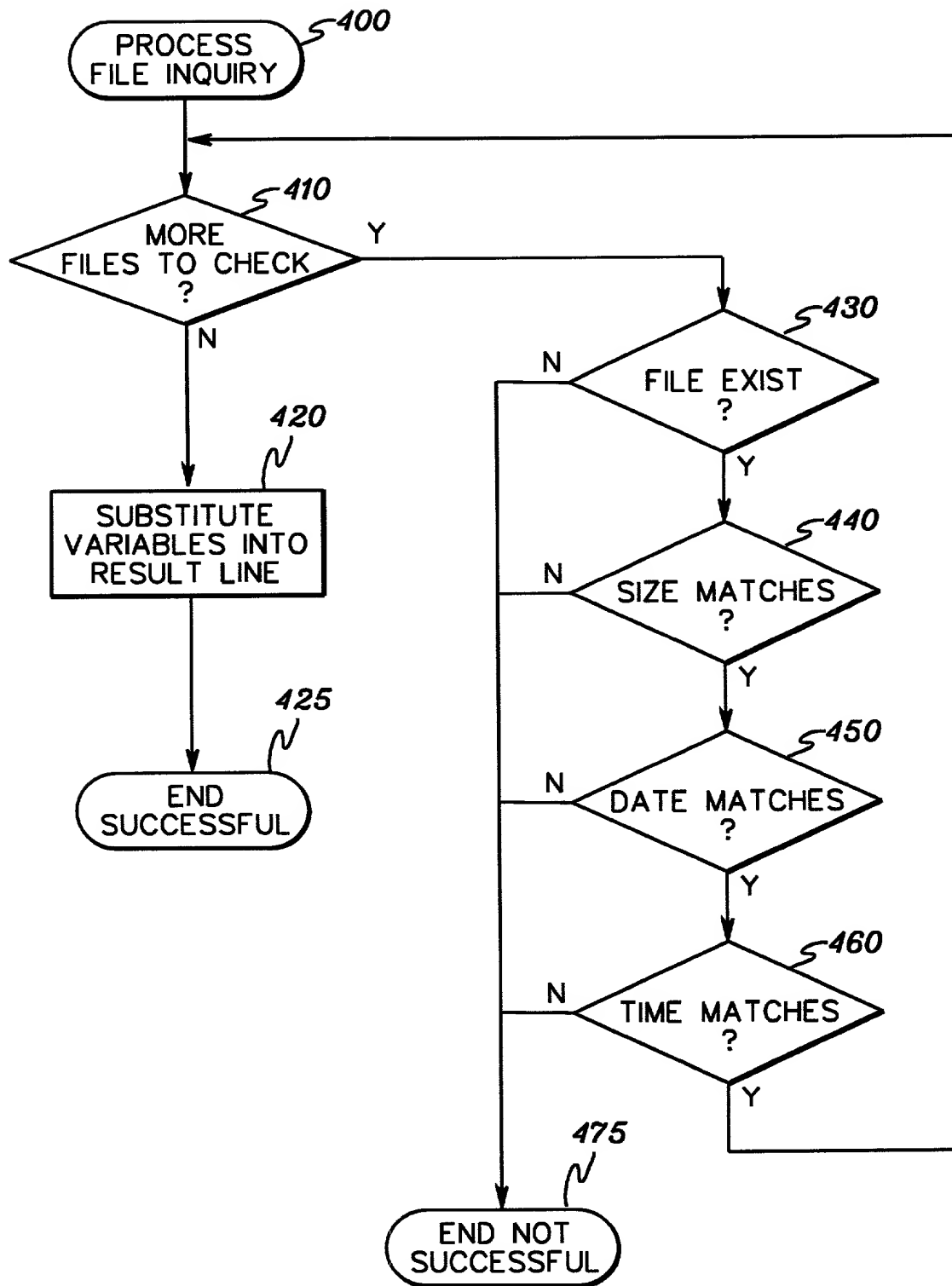
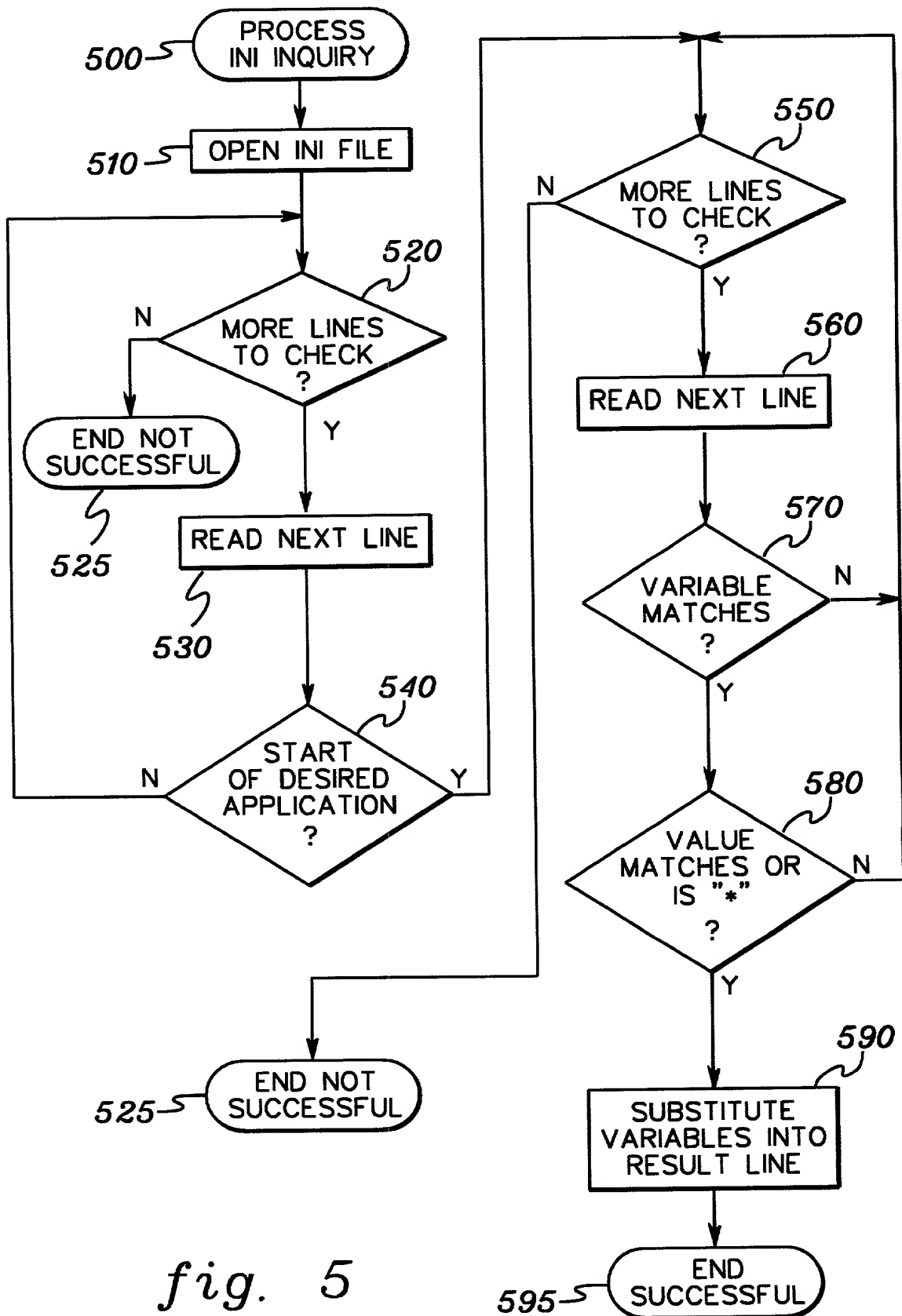
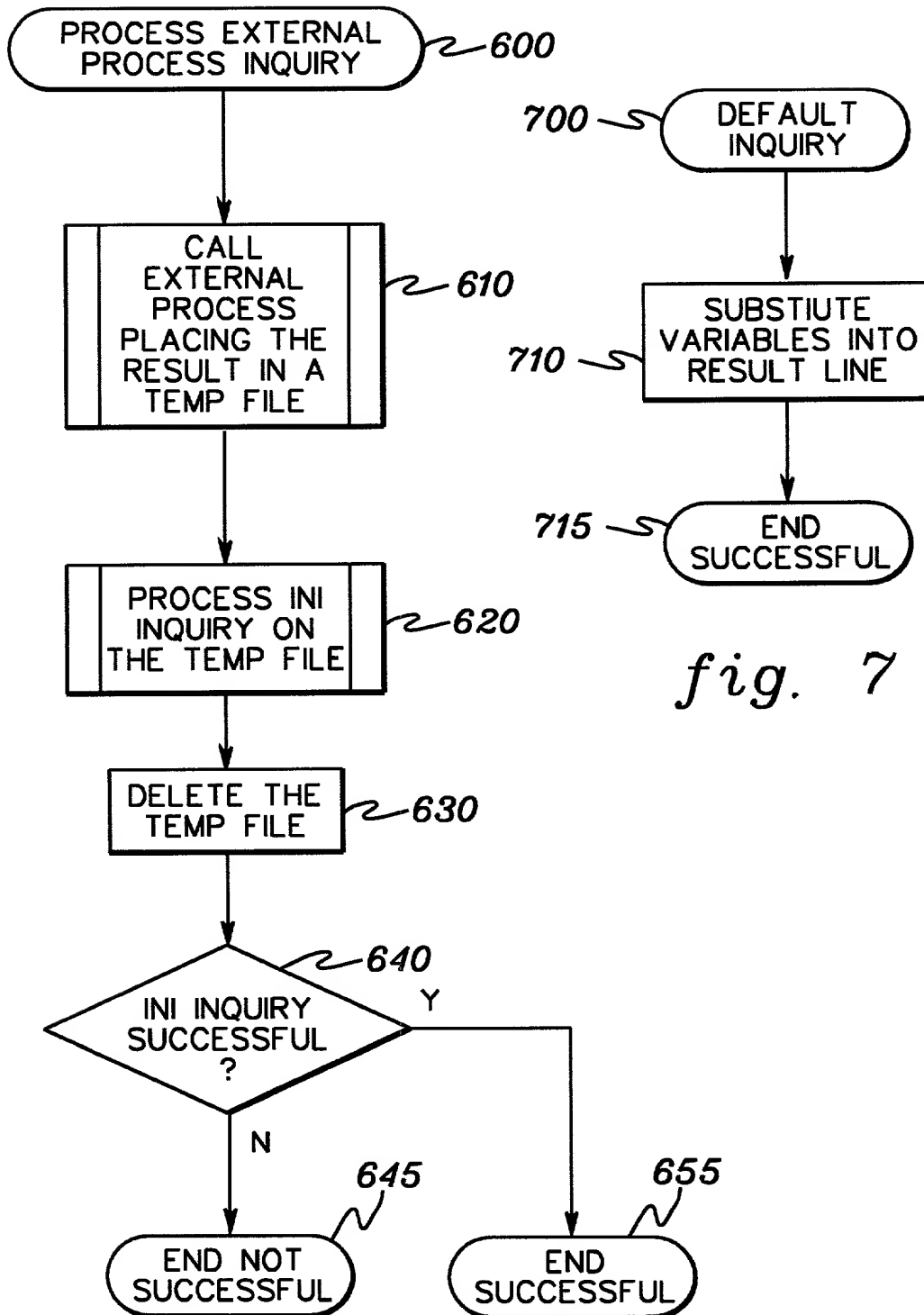


fig. 4





*fig. 7*

*fig. 6*

EXAMPLE DICTIONARY FILE

```
[IP ADDRESS]
;IPCONFIG;%APPID%=%VALUE%;IP ADDRESS;*
[HOSTNAME]
;IPCONFIG;%APPID%=%VALUE%;HOST NAME;*
[NAMESERVERS]
;IPCONFIG;%APPID%=%VALUE%;DNS SERVERS;*
[DHCP]
;IPCONFIG;DHCP IS ENABLED;DHCP ENABLED;YES
;IPCONFIG;DHCP IS NOT ENABLED;DHCP ENABLED;NO
[GATEWAY]
;IPCONFIG;%APPID%=%VALUE%;DEFAULT GATEWAY;*
[APPSW01]
;FILE;APPSW01=NOTES:4.6.1;NOTES.EXE;9/15/1997;19:15;232448
;FILE;APPSW01=NOTES:UNKNOWN %FIELD% %DATE% %TIME% %SIZE%;NOTES.EXE
[NOTESID]
;INI;NOTESID=%VALUE%;NOTES.INI;NOTES;LOCATION;*;NOTES.EXE
```

*fig. 8*



## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name. I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

INFORMATION GATHERING FACILITY EMPLOYING  
DICTIONARY FILE CONTAINING MULTIPLE INQUIRIES

the specification of which (check one)

  X   is attached hereto.

\_\_\_\_\_ was filed on \_\_\_\_\_ as United States  
Application Number or PCT International Application  
Number \_\_\_\_\_ and was amended on  
\_\_\_\_\_.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

## Priority Claimed

_____	_____	_____	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)		
_____	_____	_____	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)		

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional application(s) listed below.

_____	_____
(Application Number)	(Filing Date)
_____	_____
(Application Number)	(Filing Date)

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s), or § 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose material information as defined in 37 CFR §1.56(a) which occurred between the filing date of the prior application and the national or PCT International filing date of this application:

_____ (Appl. Serial No.)	_____ (Filing Date)	_____ (Status) (patented, pending, abandoned)
_____ (Appl. Serial No.)	_____ (Filing Date)	_____ (Status) (patented, pending, abandoned)

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Christopher A. Hughes, Esq.	Reg. No. 26,914
Edward A. Pennington, Esq.	Reg. No. 32,588
John E. Hoel, Esq.	Reg. No. 26,279
Joseph C. Redmond, Jr., Esq.	Reg. No. 18,753
David L. Adour, Esq.	Reg. No. 29,604
Lawrence R. Fraley, Esq.	Reg. No. 26,885
Arthur J. Samodovitz, Esq.	Reg. No. 31,297
William H. Steinberg, Esq.	Reg. No. 28,540
John R. Pivnichny, Ph.D.	Reg. No. 43,001
Jeff Rothenberg, Esq.	Reg. No. 26,429
Kevin P. Radigan, Esq.	Reg. No. 31,789
Blanche E. Schiller, Esq.	Reg. No. 35,670

**Send Correspondence to:**

Kevin P. Radigan, Esq.  
HESLIN & ROTHENBERG, P.C.  
5 Columbia Circle  
Albany, New York 12203-5160  
Telephone: (518) 452-5600  
Facsimile: (518) 452-5579

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

ADDED PAGE TO COMBINED DECLARATION AND POWER OF ATTORNEY  
FOR SIGNATURE BY SOLE INVENTOR

Full Name of sole inventor: Douglas George Murray

Signature: Dyl George Murray Date: 11/17/99

Residence: 241 Wyok Road, Johnson City, New York 13790

Citizenship: United States of America

Post Office Address: 241 Wyok Road, Johnson City, New York 13790